

Dokumentace k projektu z IPP, zadání SYN

Zvýraznění syntaxe

1. Analýza

V projektu bylo za úkol napsat skript v Python, který načte vstupní soubor s textem a formátovací soubor se vstupními regexama a zvoleným formátováním.

2. Řešení

2.1. Parametry

Funkce zpracovává parametry příkazové řádky a porovnává jednotlivé parametry pomocí funkce `defArg()`. Funkce zároveň hlídá všechny kolidující parametry a případně volá funkci `errors()` s kódem jedna pro ukončení skriptu s chybou.

2.2. Načítání formátovacího souboru

Pomocí funkce `format()` načítám vstup řádek po řádku a vytvářím pole polí, které obsahuje regulární výraz na prvním prvku každého pole uvnitř pole. Tenhle formát pak v `main()` upravím na dvě pole, přičemž první obsahuje pouze regulární výrazy a druhé všechny formy formátování pro daný regulární výraz. V případě formátovací chyby v jakémkoli řádku, volám funkci `errors()`. Tahle funkce však ještě neřeší správnost vstupních regulárních výrazů.

2.3. Regulární výrazy

Pomocí funkce `regex()` provádím jednoduchou syntaktickou analýzu všech vstupních regulárních výrazů. Následně taky každý vstupní regulární výraz převádím na regulární výraz použitelný v jazyce python a uložím ho na stejný index v poli, do kterého byl nahrán v předešlé funkci `format()`. Kvůli ekvivalenci indexování pole s regulárními výrazy a pole s vstupním formátováním.

2.4. Vstupní soubor

Pomocí funkce `out()` upravuju načítaný vstupní soubor podle už upravených python regulárních výrazů a formátování jim odpovídajícího. Vstupní soubor mám rozdělení znak po znaku a uložený v pomocném poli. Podle indexů z funkce `finditer()` pro jednotlivé regulární výrazy, přidávám všechna formátování z druhého pole na dané indexy v pomocném poli. V případě indexu `start()` přidám formátování před znak nacházející se na daném indexu, ale v případě `end()`, přidám formátování až za daný znak. Pak už jenom dané pole převedu zpátky na řetězec a nahraju do výstupního souboru, jenom v případě, kdy je zadán taky argument `br` dochází ještě k změně každého znaku odřádkování už ve výstupním řetězci.

3. Rozšíření

3.1. NQS

Rozšíření sem vyřešil jednoduchými regulárními výrazy na začátku funkce `format()`. V případě, že najdu jednu z kombinací ze zadání `(**, ++, +*, *+)`, upravím daný výraz na odpovídající správnému zápisu regulárního výrazu.

3.2. HTM

Rozšíření řeším pomocí funkce `regex_nooverlap()`, která kontroluje správnost formátování odpovídajícího formátování `html` kódu. V případě existence překrývání, jej upravím na formát bez překrývání.

4. Testování

Na testování sem využil v první řadě skript poskytnutý k zadání projektu a pak vlastní skript v `BASH` s přibližně třiceti testy na různé chyby či správné vstupy formátovacích a vstupních souborů.